IN THE CLAIMS

Please rewrite claims 1-14 as follows:

1. (Original) A software analysis tool comprising:

means for converting software entities and their relationships into a graph having a structure of nodes interconnected by edges, and

an editor comprising means for allowing a user to edit the graph,

wherein the graph includes a meta node and edge representing a child graph.

2. (Original) A software analysis tool as claimed in claim 1, wherein the conversion means comprises means for bi-directionally folding and unfolding a graph between meta and child levels.

3. (Original) A software analysis tool as claimed in claim 1 or 2, wherein the editor comprises means for automatically generating fresh graph layouts after manipulation.

4. (Currently Amended) A software analysis tool as claimed in claim ~~1, 2, or 3~~ 1 or 2, wherein the conversion means comprises a plurality of back-ends, each being associated with an aspect of a software system.

5. (Original) A software analysis tool as claimed in claim 4, wherein each back-end comprises means for converting the entities and the relationships of the associated aspect into nodes and edges of the graph.

6. (Currently Amended) A software analysis tool as claimed in ~~claims 4 or 5~~ claim 4, wherein the back-ends are associated with managers.

7. (Original) A software analysis tool as claimed in claim 6, wherein the managers comprise means for routing commands between the editor and the back-ends.

8. (Currently Amended) A software analysis tool as claimed in ~~claims 6 or 7~~ claim 6, wherein each manager is associated with a group of back-ends associated with a group of back-ends.

9. (Original) A software analysis tool as claimed in claim 8, wherein the back-ends associated with a particular manager share a common interface and set of operations.

10. (Currently Amended) A software analysis tool substantially as described with reference to ~~the drawings~~ Figs. 1-18.

11. (Original) A dependency analysis system recorded on a computer-readable medium, comprising:

a node class for instiating node objects in memory representing aspects of an analyzed system as nodes of a graph;

a connection class for instantiating connection objects in memory representing dependencies between aspects of an analyzed system;

an edge class for instantiating edge objects representing collections of one or more connections or edges.

12. (Original) The dependency analysis system of claim 11, further comprising:

at least one subclass of the node class, the subclass being specific to a particular category of system.

13. (Original) A dependency analysis system recorded on a computer-readable medium, comprising:

an abstraction layer for providing a uniform interface to third-party analysis tools;

a graph model data structure for storing dependency information derived through the abstraction layer from third-party tools;

a rendering system for providing a plurality of views of the graph model data structure.

14. (Original) A dependency analysis system comprising:

a data structure stored in computer memory representing a hierarchy of graphs;

a rendering system for displaying the hierarchy of graphs;

a user interface responsive to a user action indicating a command to expand a displayed node, the user interface causing the rendering system to replace the displayed node with one or more child nodes in response to the user action.